Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (Previously presented): A process for registering data in a data management system and identifying uses of the registered data by users, comprising:

creating a key template comprising a plurality of elements, wherein each element is defined by an element size, a quantity, a start position and an initial position;

receiving a source file from data owners;

creating a fingerprint for the source file by recording portions of the source file that correspond to each of the elements in the key template;

storing the source file and fingerprint in a database; and

comparing unknown data files to the fingerprint stored in the database to determine whether the unknown data files are copies of any portion of the source file.

Claim 2 (Previously Presented): A process as claimed in claim 1, further comprising branding the source file, wherein the branding is associated with indicia of ownership.

Claim 3 (Original): A process as claimed in claim 2, wherein branding comprises:

receiving a user defined data block from a user, wherein the data block includes user defined information;

examining the source file to determine whether the source file contains a data block;

building a concatenated string from the data block information;

embedding the data block within the source file.

Claim 4 (Previously Presented): A process as claimed in claim 3, wherein the user defined information contained in the data block includes any of the following: rights information, licencing information, a counter, key words, file attributes and mandatory compliance information.

Claim 5 (Original): A process as claimed in claim 4, wherein compliance information comprises any of the following group: identification, age information, custodial information and other mandatory information required by law for image data.

Claim 6 (Original): A process as claimed in claim 3, further comprising:

verifying whether the source file currently exists in the system;

creating a fingerprint for the source file if the file is not stored in the data management database; and

storing the source file and the associated file fingerprint in the database.

Claim 7 (Original): A process as claimed in claim 2, wherein branding further comprises:

receiving a request to brand a source file from a user;

retrieving a preassigned encryption key for the user, wherein the encryption key is stored in the database in association with the source file;

verifying that the user requesting the branding of the source file is authorized to request the branding of the file;

rejecting the branding and notifying the file owner if the requesting user is not authorized to brand the file;

if the requesting user is authorized to request the branding, encrypting the data block utilizing the preassigned encryption key assigned to the user; and

embedding the encrypted data block into the source file; and

creating a fingerprint of the source file with the embedded data block.

Claim 8 (Original): A process as claimed in claim 1, wherein the data includes pixel values and a plurality of color values for each pixel, and wherein creating a fingerprint further comprises:

averaging color values for predefined portions of the source file.

Claim 9 (Original): A data management system for managing, reviewing, comparing and detecting data on a network, comprising:

a data management server;

a key generator;

a source print generator; and

a source print detector.

Claim 10 (Original): A data management system as claimed in claim 9, further comprising a data embedding system.

Claim 11 (Original): A data management system as claimed in claim 9, the source print detector further comprising:

a searching member, wherein the searching downloads unknown files to the data management server from the network; and

a comparison member, the comparison member includes a storage database, and is configured to review the unknown files and compare the unknown files to a set of source prints stored on the storage database.

Claim 12 (Previously Presented): A process of registering, monitoring and tracking uses of data registered in a data management system on a network, comprising:

creating a key template comprising a plurality of elements, wherein each element is defined by an element size, a quantity, a start position and an initial position;

receiving a source file from data owners;

creating a fingerprint file for the source file by recording portions of the source file that correspond to each of the elements in the key template;

storing the source file and fingerprint in a database;

searching the network for unknown files;

downloading unknown files to a data management server;

recording portions of the unknown files that correspond to each of the elements in the key template to create a fingerprint for the unknown file;

comparing the fingerprint of the unknown file to the fingerprint of the source file; and

assigning a probability matching level for the unknown file based upon the comparison results of the comparison between the fingerprint of the unknown file and the fingerprint of the source file.

Claim 13 (Previously Presented): A method of providing a comparison between elements in a source file and elements in a target file, including the steps of:

providing a fingerprint of the source file, the fingerprint including a plurality of elements in the source file,

transferring the fingerprint into a data management server,

retrieving the fingerprint from the data management server when it is desired to compare the elements in the target file with the fingerprint in the source file, and

comparing the fingerprint with corresponding elements in the target file to determine if there is a coincidence between the source file and the target file.

Claim 14 (Previously Presented): A method as set forth in claim 13, further including the steps of:

branding the source file with indicia of ownership, and

determining if the target file is branded with the indicia of ownership in the source file.

Claim 15 (Previously Presented): A method as set forth in claim 13, wherein

a collection of statistical samples are obtained from the source file to generate for the source file a unique identifier which can be search for in the target file.

Claim 16 (Previously Presented): A method as set forth in claim 13, including the steps of:

determining the size of the fingerprint and the number of elements in the fingerprint and the size of the elements in the fingerprint and the characteristic of the elements in the fingerprint, and

using these determinations to decide whether the target file has a coincidence.

Claim 17 (Previously Presented): A method as set forth in claim 13, wherein

the server comprises a programmable processor.

Claim 18 (Previously Presented): A method as set forth in claim 14, wherein

a collection of statistical samples are obtained from the source file to generate for the source file a unique identifier which can be searched for in the target file and wherein the size of the fingerprint and the number of elements in the fingerprint and the size of the elements in the fingerprint and the characteristics of the elements in the fingerprint are determined and wherein

the server comprises a programmable processor.

Claim 19 (Previously Presented): A method of providing a comparison between a source file and a target file, including the steps of:

providing a data management system including a data management server for holding the source file,

a key generator in the data management server that identifies a plurality of elements in the source file to be compared with elements in the target file, the key generator including a key defining the position of the elements in the source file to be compared with elements in the target file,

extracting the key elements in the source file from the data management server, and

comparing the extracted elements from the source file with the corresponding elements in the target file to determine if there is a coincidence between the elements in the source file and the elements in the target file.

Claim 20 (Previously Presented): A method as set forth in claim 19, including the steps of:

providing a start position for the key and providing initial positions for different elements in the key, and

using the start position and the initial positions of the different elements in the key to locate a start position and initial positions in corresponding elements in the target file.

Claim 21 (Previously Presented): A method as set forth in claim 19, including the steps of:

normalizing the source file before identifying the elements in the key.

Claim 22 (Previously Presented): A method as set forth in claim 19, including the step of:

adjusting the rate of sampling the elements in the target file to the rate of sampling the elements in the key in the source file.

Claim 23 (Previously Presented): A method as set forth in claim 20, including the steps of:

normalizing the source file before identifying the elements in the key, and

adjusting the rate of sampling the elements in the target file to the rate of sampling the elements in the key in the source file.

Claim 24 (Previously Presented): A method of providing a comparison between a source file and a target file, including the steps of:

providing a data management system including a data management server,

providing a source file in the data management server,

providing a key in the source file,

providing seeds to facilitate the searching of files to determine if they are target files,

downloading files to the data management server when it is determined that they are target files, and

initiating the comparison of elements in the key of the source file with corresponding elements in the target file when the target file is downloaded to the data management server.

Claim 25 (Previously Presented): A method as set forth in claim 24, including the step of:

defining the elements in the key by quantifiers, including the size of the elements in the key, the quantity of the elements in the key and the initial positions of the elements in the key and the types of the elements in the key.

Claim 26 (Previously Presented): A method as set forth in claim 24, including the step of:

extracting the elements of the key by a selected one of a compression specific method and a non-compression specific method.

Claim 27 (Previously Presented): A method as set forth in claim 26, wherein

the compression specific method provides for a copying of data within the source file in a compressed format.

Claim 28 (Previously Presented): A method as set forth in claim 26, wherein the non-compression specific method involves a decompression of compressed data in the source file.

Claim 29 (Previously Presented): A method as set forth in claim 26 wherein

The data in the source file is normalized prior to the extraction of the data for the key elements from the source file and wherein

The compression specific method provides for a copying of the data within the source file in a compressed format and wherein

The non-compression specific method involves a decompression of compressed data in the source file.

Claim 30 (Previously Presented): A method as set forth in claim 24 wherein

the data in the source file is normalized prior to the extraction of the data for the key elements from the source file and wherein

the elements in the key are defined by quantifiers, including the size of the elements in the key, the quantity of the elements in the key and the initial positions of the elements in the key and the types of the elements in the key and wherein

the compression specific method provides for a copying of the data within the storage file and is applied to data in a compressed format

the non-compression method involves a decompression of data in the source file.

Claim 31 (Previously Presented): A method of providing a comparison between a source file and a target file, including the steps of:

providing a key in the source file,

providing a start position in the source file for the key,

providing a seed to facilitate the search of files to determine if they have a key corresponding to the key in the source file,

providing a data management system including a data management server for storing the source file,

downloading a searched file to the data management server as the target file when it is determined that the searched file has a key corresponding to the key in the source file, determining the start position in the target file from the start position in the key, and

comparing elements in the target file with the elements in the key in the source file to determine if there is a coincidence between the source file and the target file.

Claim 32 (Previously Presented): A method as set forth in claim 31, including the steps of:

defining elements in the key, each of the elements begin defined by an initializing position spaced from the start position by a particular distance in the source file, and

determining if there is a coincidence between the source file and the target file in accordance with the positioning of the elements in the target file relative to corresponding elements in the source file.

Claim 33 (Previously Presented): A method as set forth in claim 31, including the steps of:

defining the lengths of elements in the key in the source file by the distance of the elements from the initial positions of the elements in the key, and

determining if there is a coincidence between the source file and the target file in accordance with the lengths of the elements in the target file relative to the lengths of corresponding elements in the key in the target file.

Claim 34 (Previously Presented): A method as set forth in claim 31, including the steps of:

defining the sequence of the elements in the key in the source file, and

determining the sequence of the elements in the target file relative to the sequence of the elements in the key in the source file to decide if there is a coincidence between the target file and the source file.

Claim 35 (Previously Presented): A method as set forth in claim 31, wherein

the source file is normalized before any determination of a coincidence between the elements in the target file and the elements in the source file.

Claim 36 (Previously Presented): A method as set forth in claim 31, including the steps of:

defining elements in the key, each of the elements being defined by the distance of an initializing position in the element from the start position in the key in the source file, determining if there is a coincidence between the source file and the target file in accordance with the positioning of the elements in the target file relative to the elements in the source file,

defining the lengths of elements in the key in the source file by the lengths of the elements from the initial positions of the elements in the key,

determining if there is a coincidence between the source file and the target file in accordance with the lengths of the elements in the target file relative to the lengths of corresponding elements in the key in the source file,

defining the sequence of the elements in the key in the source file, and

determining the sequence of the elements in the target file relative to the sequence of the elements in the key in the source file to decide if there is a coincidence between the elements in the target file and the elements in the source file, and

normalizing the source file before any comparison between the elements in the target file and the elements in the source file.

Claim 37 (Previously Presented): A method of providing a comparison between a source file and a target file, including the steps of:

providing a data management system including a data management server,

recording a source file in the data management server,

providing a predefined key in the source file to create a fingerprint, the key being defined by a set of elements in the source file,

storing the key in a database,

comparing elements in the target file with the elements in the key in the database to determine a coincidence between the elements in the key in the source file and the elements in the target file, and

making a record of the target file when at least a particular match occurs between the elements in the key in the source file and the elements in the target file.

Claim 38 (Previously Presented): A method as set forth in claim 37, including the steps of:

branding the source file,

comparing the branding of the source file with any branding in the target file, and identifying the target file when the comparison indicates that there is a branding in the target file corresponding to the branding in the source file.

Claim 39 (Previously Presented): A method as set forth in claim 37 wherein

the branding of the source file relates to individual ones of rights information, licensing information, a counter, key words, user defined attributes and mandatory compliance information.

Claim 40 (Previously Presented): A method as set forth in claim 37, including the steps of:

providing a data block for the source file,

encrypting the data block in the source file,

providing a decryption key that is associated with an authorized user of the source file, and

using the data block and the decryption key to verify authorized use of the source file by the user.

Claim 41 (Previously Presented): A method as set forth in claim 37, including the steps of:

providing a seed to facilitate the search of files to constitute target files,

providing a data management system including a data management server for storing the source file,

downloading a searched file to the data management server when it is determined that the searched file is a target file,

providing a key in the source file,

defining a start position in the source file for the key,

defining a plurality of elements in the key,

determining the start position in the target file from the start position in the key, and

comparing elements in the target file with elements in the key to determine if at least a particular match exists between the elements in the key and the elements in the target file.

Claim 42 (Previously Presented): A method of generating a key for a source file, including the steps of:

providing a data management system including a data management server,

providing a storage file in the data management server,

selecting a plurality of elements in the source file to define a key,

each of the elements being defined by a quantifier of the key,

the quantifier for each of the elements including element size and initial positions of the elements,

inputting the quantifier into the data management server for each of the elements in the key, and

defining a start position from which the commencement of the key begins.

Claim 43 (Previously Presented): A method as set forth in claim 42 wherein

the type of each key element identifies the nature of the information in the key element and wherein

the type of each key element includes individual ones of bits, bytes, file segments and paragraphs.

Claim 44 (Previously Presented): A method as set forth in claim 42, including the step of:

identifying each element in the key by an initial position of the element relative to a start position from which the commencement of the key begins.

Claim 45 (Previously Presented): A method as set forth in claim 42, including

providing a plurality of keys in a storage file, and

providing changes in individual ones of the keys in the storage file.

Claim 46 (Previously Presented): A method as set forth in claim 42, including

identifying each element in the key by an initial position of each element relative to the start position from which the commencement of the key begins,

providing a plurality of keys in a storage file, and

providing changes in individual ones of the keys in the storage file wherein

the type of each key element identifies the nature of the information in the key element and includes individual ones of bits, bytes, file segments and paragraphs.

Claim 47 (Previously Presented): A method of providing a key for a source file, including the steps of:

providing a data management system including a data management server,

providing a source file in the data management server,

selecting a plurality of elements in the source file to define a key,

providing a start position for the key,

providing characteristics for each of the elements in the key including an initializing position of the element relative to the start position in the key, and

inputting into the data management server the key including the start position, the elements and the characteristics for each of the elements in the key including the initializing position of the element relative to the start position in the key.

Claim 48 (Previously Presented): A method as set forth in claim 46, including the steps of:

each of the elements in the key including data,

normalizing the data in each of the elements in the key before inputting the elements in the key into the source file, and

extracting the normalized data in the key elements.

Claim 49 (Previously Presented): A method as set forth in claim 46, including the steps of:

seeding a target file to provide a searching start for targeting specific areas of interest in the target file,

searching the target file for the specific areas of interest represented by the seeding, and

comparing the specific areas of interest in the target file with the elements in the key in the source file to determine if the target file corresponds to the source file.

Claim 50 (Previously Presented): A method as set forth in claim 47, including the steps of:

providing quantifiers for the elements in the source file wherein,

the comparison is based upon the existence in the target file of elements in the key and the order of the elements in the target file relative to the order of elements in the key and the quantifiers of the elements in the key relative to the quantifiers of elements in the target file.

Claim 51 (Previously Presented): A method as set forth in claim 47, including the steps of:

each of the elements in the key including data,

normalizing the data in each of the elements in the key before inputting the element in the key into the source file, and

extracting the normalized data in the key elements,

seeding the target file to provide a searching start for targeting specific areas of interest in the target file,

searching the target file for the specific areas of interest represented by the seeding,

providing quantifiers for the elements in the source file, and

comparing the specific areas of interest in the target file with the elements in the key in the source file to determine if the elements in the target file correspond to the elements in the source file and wherein

the comparison is based upon the existence in the target file of elements in the key and the order of the elements in the target file relative to the order of the elements in the key and the quantifiers of the elements in the target file relative to the quantifiers of the elements in the key.

Claim 52 (Previously Presented): A method of reviewing data in a target file relative to data in a source file, including the steps of:

providing a data management system including a data management server for controlling the data in the source file,

generating a key comprising a plurality of elements in the source file,

inputting the key including the plurality of elements into the data management server,

applying the key to the elements in the target file, and

extracting from the target file the elements defined by the key.

Claim 53 (Previously Presented): A method as set forth in claim 51 wherein

information relating to the elements in the key include the number of the elements in the key, the sizes of the elements in the key, the start position of the key and the initializing positions of the elements in the key and wherein the elements are extracted from the target file in accordance with the number of elements in the key, the sizes of the elements in the key, the start position of the key and the initializing positions of the elements in the key.

Claim 54 (Previously Presented): A method as set forth in claim 51 wherein

the elements extracted from the target file are compared with the elements in the key to determine if the elements in the target file correspond to the elements in the key.

Claim 55 (Previously Presented): A method as set forth in claim 53, including the steps of:

defining the key by the number of elements in the key, the sizes of the elements in the key, the start position of the key and the initial position of the elements in the key, and

extracting the elements from the target file in accordance with the number of elements in the key, the sizes of the elements in the key, the start position of the key and the initial positions of the elements in the key.

Claim 56 (Previously Presented): A method as set forth in claim 55 wherein

the key elements are extracted from the source file by either a compression specific method or a non-compression specific method and wherein

the compression specific method provides for a copying of data within the source file in a compressed format and wherein

the non-compression specific method involves a decompression of compressed data in the source file.

Claim 57 (Previously Presented): A method of authenticating data in a target file relative to data in a source file, including the steps of:

providing data in the source file in a storage medium,

providing in the source file a data block branding the data in the source file,

determining if the target file includes a data block branding the data in the target file, and

comparing the data blocks in the source file and in the target file when the target file includes a data block branding the data in the target file.

Claim 58 (Previously Presented): A method as set forth in claim 57 wherein

when there is a coincidence between the data blocks in the source file and the target file, determining if the target file has been authorized by the owner of the source file to provide the data block in the target file.

Claim 59 (Previously Presented): A method as set forth in claim 57 wherein

the data block in the source file contains information pertaining to intellectual property.

Claim 60 (Previously Presented): A method as set forth in claim 57 wherein

the data block in the source file contains mandatory compliance information.

Claim 61 (Previously Presented): A method as set forth in claim 57 wherein

the data block in the source file includes use rights.

Claim 62 (Previously Presented): A method as set forth in claim 57 wherein

when there is a coincidence between the data block in the source file and the data block in the target file, determining if the target file has been authorized by the owner of the source file to provide the data block in the target file and wherein

the data block in the source file contains information pertaining to individual ones of intellectual property rights.

Claim 63 (Previously Presented): A method of providing a key in a source file containing data, including the steps of:

providing a data management system including a data management server for controlling the data in the source file,

providing a key in the source file, the key including a plurality of elements in the source file,

providing a start position for the key and initializing positions for the elements in the key, and inputting into the data management server the key including the plurality of elements, the start position for the key and the initializing position for the elements in the key.

Claim 64 (Previously Presented): A method as set forth in claim 62 wherein

the elements in the key are extracted from the key in the data management server.

Claim 65 (Previously Presented): A method as set forth in claim 62 wherein

the elements in the key are selected from a group including bits, bytes, file segments and paragraphs.

Claim 66 (Previously Presented): A method as set forth in claim 62 wherein

applying the key in the source file in the data management server to the target file to recover from the target file elements defined by the key.

Claim 67 (Previously Presented): A method as set forth in claim 62 wherein

the elements in the key are compared with the elements recovered from the target file to determine if the elements recovered from the target file correspond to the elements in the key.

Claim 68 (Previously Presented): A method of providing a plurality of keys in a storage file containing data, including the steps of:

providing a data management system including a data management server for controlling the data in a source file,

providing a plurality of keys each including a plurality of elements in the source file,

providing a start position for each of the keys and initializing positions for the elements in each of the keys,

inputting into the data management server each of the keys in the source file, including the plurality of elements in the key, the start position for the key and the initializing positions for the elements in the key, and

selectively adding, eliminating and changing keys in the source file in the data management server.

Claim 69 (Previously Presented): A method as set forth in claim 68 wherein

the elements in the keys are extracted from the keys in the data management server.

Claim 70 (Previously Presented): A method as set forth in claim 68 wherein

the elements in the keys are selected from groups including bits, bytes, file segments and paragraphs.

Claim 71 (Previously Presented): A method as set forth in claim 68, including the steps of:

applying the keys in the source file in the data management server to the target file to recover from the target file elements defined by the elements in the keys.

Claim 72 (Previously Presented): A method as set forth in claim 68 wherein

the elements in the keys are compared with the recovered from the target file to determine if the elements recovered from the target file can be considered to be a copy of the elements in the keys.

Claim 73 (Previously Presented): A method as set forth in claim 68 wherein

the elements in the keys are extracted from the key in the data management server and wherein

the elements in the keys are selected from groups including bits, bytes, file segments and paragraphs and wherein

the keys in the source file in the data management server are applied to the target file to recover from the target file elements defined by the elements in the keys and wherein

the elements in the keys are compared with the elements recovered from the target file to determine if the elements recovered from the target file can be considered to be a copy of the elements in the keys.

Claim 74 (Previously Presented): A method as set forth in claim 68, including the step of:

normalizing the elements in the keys.

Claim 75 (Previously Presented): A method as set forth in claim 68, including the steps of:

extracting the elements from the keys by a selected one of a compression selective method and a non-compression selective method and wherein

the compression selective method provides for a copying of data in a compressed format in the source file and wherein

the non-compression selective method provides for a decompression of compressed data in the source file.

Claim 76 (Previously Presented): A method of providing a compression between a source file and a target file, including the steps of:

providing a data block for the source file,

encrypting at least a portion of the data in the source file,

providing a decryption key that is associated with an authorized user of the source file and that decrypts the encryption in the data block, and

using the data block and the decryption key to verify an authorized use of the source file by the user.

Claim 77 (Previously Presented): A method as set forth in claim 76 wherein

a portion of the data block is encrypted and another portion of the data is not encrypted such that the unencrypted portion of the data block is available to any user reviewing the source file.

Claim 78 (Previously Presented): A method as set forth in claim 76 wherein

the data block comprises a set of user defined information that is encrypted, at least in part, utilizing an algorithm assigned to an authorized user.

Claim 79 (Previously Presented): A method as set forth in claim 78 wherein

the encrypted information is embedded into the data block and wherein

only users receiving the decryption key can decrypt the encrypted information embedded into the data block.

Claim 80 (Previously Presented): A method as set forth in claim 76 wherein

a portion of the data block is decrypted and another portion of the data is not encrypted such that the unencrypted portion of the data block is available to any user reviewing the source file and wherein

the data block comprises a set of user defined information that is encrypted, at least in part, utilizing an algorithm assigned to an authorized user and wherein

the encrypted information is embedded into the data block and wherein

only users receiving the decryption key can decrypt the encrypted information embedded into the data block.

Claim 81 (Previously Presented): A process as claimed in claim 3, wherein

the user defined information contained in the data block includes a licensing counter for tracking the number of times a file has been licensed for use.

Claim 82 (Previously Presented): A method for fingerprinting a data file registered in a data management system on a network, comprising the steps of:

defining a fingerprint key, comprising the steps of:

selecting a sample quantity parameter, wherein the sample quantity parameter is equal to a number of samples to be taken from the data file;

selecting at least one sample length parameter, wherein the sample length parameter is associated with a length of samples to be taken from the data file;

selecting a start location, wherein the start location is associated with at least one of a physical beginning of the data file or a logical beginning of the data file;

selecting a relative position parameter, wherein the relative position parameter is associated with a location from which samples are to be taken, relative to the start location;

selecting a plurality of predetermined sampling rules;

extracting a plurality of fingerprint elements from the data file using the defined fingerprint key; and

concatenating the plurality of fingerprint elements, wherein a fingerprint that uniquely identifies the data file is created.